

What is claimed is:

1. In a computer system having a plurality of processors connected to a shared memory, a method of decoupling an address from write data in a store to the shared memory, comprising:

generating a write request address for a memory write, wherein the write request address points to a memory location in shared memory;

issuing a write request to the shared memory, wherein the write request includes the write request address;

noting the write request address in the shared memory;

comparing, in the shared memory, addresses in subsequent load and store requests to the write request address;

transferring the write data to the shared memory;

matching, within the shared memory, the write request address to the write data; and

storing the write data into the shared memory as a function of the write request address.

2. The method according to claim 1, wherein the shared memory includes a store address buffer and wherein noting the write request address includes writing the address in the store address buffer.

3. The method according to claim 2, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

4. The method according to claim 1, wherein the shared memory includes a cache, wherein noting the write request address includes changing a state in a cache line associated with the write request address to “WaitForData”, and wherein comparing

addresses in subsequent load and store requests to the write request address includes accessing the cache and stalling if a cache line hit returns a “WaitForData” state.

5. The method according to claim 1, wherein the shared memory includes a bit vector, wherein noting the write request address in the shared memory includes setting one or more bits in the bit vector corresponding to the write request address, and wherein comparing addresses in subsequent load and store requests to the write request address includes comparing bits that would be set corresponding to the load and store request addresses the bits set for the write request address and stalling servicing of the load and store requests if there is a match.

6. The method according to claim 1, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

7. The method according to claim 6, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other than the write request address without waiting for the write data to be written to the write request address.

8. The method according to claim 1, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other than the write request address without waiting for the write data to be written to the write request address.

9. The method according to claim 1, wherein comparing addresses in subsequent read and write requests includes enforcing memory ordering in subsequent read and write requests to the write request address until the write data associated with the first write request is written into the shared memory.

10. The method according to claim 1, wherein issuing a write request includes ensuring that all vector and scalar loads from shared memory for that processor have been sent to shared memory prior to issuing the write request.

11. In a computer system having a plurality of processors connected to a shared memory, a method of decoupling an address from write data in a write to the shared memory, comprising:

generating a write request address for a memory write, wherein the write request address points to a memory location in shared memory;

issuing a first write request to the shared memory, wherein the first write request includes the write request address;

noting the write request address in the shared memory;

comparing, in the shared memory, addresses in subsequent read and write requests to the write request address;

stalling subsequent read requests to the write request address until the write data is written into the shared memory; and

if the address in a subsequent write request matches the write request address stored in shared memory and there are no stalled read requests to the write request address, discarding the first write request.

12. The method according to claim 11, wherein the shared memory includes a store address buffer and wherein noting the write request address includes writing the address in the store address buffer.

13. The method according to claim 12, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

14. The method according to claim 11, wherein the shared memory includes a cache, wherein noting the write request address includes changing a state in a cache line associated with the write request address to “WaitForData”, and wherein comparing addresses in subsequent load and store requests to the write request address includes accessing the cache and stalling if a cache line hit returns a “WaitForData” state.

15. The method according to claim 11, wherein the shared memory includes a bit vector, wherein noting the write request address in the shared memory includes setting one or more bits in the bit vector corresponding to the write request address, and wherein comparing addresses in subsequent load and store requests to the write request address includes comparing bits that would be set corresponding to the load and store request addresses the bits set for the write request address and stalling servicing of the load and store requests if there is a match.

16. The method according to claim 11, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

17. The method according to claim 16, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other than the write request address without waiting for the write data to be written to the write request address.

18. The method according to claim 11, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other than the write request address without waiting for the write data to be written to the write request address.

19. The method according to claim 11, wherein comparing addresses in subsequent read and write requests includes enforcing memory ordering in subsequent read and write requests to the write request address until the write data associated with the first write request is written into the shared memory.

20. The method according to claim 11, wherein issuing a write request includes ensuring that all vector and scalar loads from shared memory for that processor have been sent to shared memory prior to issuing the write request.

21. In a computer system having a plurality of processors connected to a shared memory, a method of decoupling an address from write data in a store to the shared memory, comprising:

generating a write request address for a vector store to memory, wherein the write request address points to a memory location in shared memory;

issuing a vector store request to the shared memory, wherein the write request includes the write request address;

noting the write request address in the shared memory;

comparing, in the shared memory, addresses in subsequent load and store requests to the write request address;

transferring the write data from a vector register to the shared memory;

matching, within the shared memory, the write request address to the write data;

and

storing the write data into the shared memory as a function of the write request address.

22. The method according to claim 21, wherein the shared memory includes a store address buffer and wherein noting the write request address includes writing the address in the store address buffer.

23. The method according to claim 22, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

24. The method according to claim 21, wherein the shared memory includes a cache, wherein noting the write request address includes changing a state in a cache line associated with the write request address to “WaitForData”, and wherein comparing addresses in subsequent load and store requests to the write request address includes accessing the cache and stalling if a cache line hit returns a “WaitForData” state.

25. The method according to claim 21, wherein the shared memory includes a bit vector, wherein noting the write request address in the shared memory includes setting one or more bits in the bit vector corresponding to the write request address, and wherein comparing addresses in subsequent load and store requests to the write request address includes comparing bits that would be set corresponding to the load and store request addresses the bits set for the write request address and stalling servicing of the load and store requests if there is a match.

26. The method according to claim 21, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

27. The method according to claim 26, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other than the write request address without waiting for the write data to be written to the write request address.

28. The method according to claim 21, wherein comparing addresses in subsequent read and write requests includes servicing the load and store requests to addresses other

than the write request address without waiting for the write data to be written to the write request address.

29. The method according to claim 21, wherein comparing addresses in subsequent read and write requests includes enforcing memory ordering in subsequent read and write requests to the write request address until the write data associated with the first write request is written into the shared memory.

30. The method according to claim 21, wherein issuing a write request includes ensuring that all vector and scalar loads from shared memory for that processor have been sent to shared memory prior to issuing the write request.

31. A method of decoupling vector data stores from vector instruction execution, comprising:

executing a vector instruction on vector data stored in a vector register, wherein executing a vector instruction includes storing result vector data in a vector register;

generating a vector write address for a vector store;

issuing a vector store request to memory, wherein the vector store request includes the vector write address;

transferring result vector data from the vector register to memory;

matching the vector store request and result vector data in memory; and

storing the result vector data into memory as a function of the address in the vector store request.

32. The method according to claim 31, wherein matching includes comparing addresses in subsequent read and write requests to the vector write address and stalling subsequent read requests to the vector write address until the result vector data is written into the memory.

33. The method according to claim 31, wherein matching includes comparing addresses in subsequent read and write requests received from other processing units to the vector write address and stalling subsequent read requests to the vector write address until the result vector data is written into the memory.

34. In a processor having a plurality of processing units connected to a shared memory, a method of decoupling an address from write data in a write to the shared memory, comprising:

generating a write request address for a memory write, wherein the write request address points to a memory location in shared memory;

issuing a write request to the shared memory, wherein the write request includes the write request address;

storing the write request address in the shared memory;

comparing addresses in subsequent read and write requests to the write request address stored in shared memory;

transferring the write data to the shared memory;

matching, within the shared memory, the write request address to the write data; and

storing the write data into the shared memory as a function of the write request address.

35. The method according to claim 34, wherein issuing a write request includes ensuring that all vector and scalar loads from shared memory for that processor have been sent to shared memory prior to issuing the write request.

36. The method according to claim 34, wherein comparing addresses in subsequent read and write requests includes stalling subsequent read requests to the write request address until the write data is written into the shared memory.

37. The method according to claim 34, wherein comparing addresses in subsequent read and write requests includes enforcing memory ordering in subsequent read and write requests to the write request address until the write data associated with the first write request is written into the shared memory.

38. A computer system, comprising:

    a plurality of processors, wherein the processors includes means for issuing a write address separate from data to be written to the write address; and

    a shared memory connected to the plurality of processors, wherein the shared memory includes:

        means for receiving a write request including a write address; and

        means for stalling subsequent loads and stores to the write address in shared memory until the data to be written to the write address is received and written by the shared memory.